**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



Hình 1 : Hình logo Đại Học Thủy Lợi

Nhóm sinh viên thực hiện:

1. Đào Đăng Tài - 2151264682, 63TTNT
2. Nguyễn Văn Qúy - 2151264679 63TTNT
3. Doãn Phúc Được - 2151264652, 63TTNT
4. Nguyễn Hữu Tiến - 2151264687,63TTNT

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN TIỀN XỬ LÝ DỮ LIỆU**

ĐỀ TÀI: DỰ ĐOÁN KẾT QUẢ BÓNG ĐÁ

Giáo viên hướng dẫn: TS. Tạ Quang Chiểu

Hà Nội, năm 2024

**MỤC LỤC**

[PHẦN 1. GIỚI THIỆU BÀI TOÁN 3](#_Toc162778529)

[PHẦN 2. TIỀN XỬ LÝ DỮ LIỆU 5](#_Toc162778530)

[2.1. Xử lý dữ liệu thiếu 5](#_Toc162778531)

[2.2. Mã hóa dữ liệu 6](#_Toc162778532)

[2.3. Xử lý dữ liệu nhiễu 7](#_Toc162778533)

[2.4. Xử lý giá trị hỗn hợp 10](#_Toc162778534)

[2.5. Xử lý mất cân bằng dữ liệu 12](#_Toc162778535)

[2.6. Feature Scaling 14](#_Toc162778536)

[PHẦN 3. ỨNG DỤNG MÔ HÌNH HỌC MÁY ĐỂ THỰC HIỆN VỚI BÀI TOÁN 14](#_Toc162778537)

[3.1. Chia dữ liệu 14](#_Toc162778538)

[3.2. Huấn luyện mô hình 15](#_Toc162778539)

[3.3. Kiểm tra độ chính xác của mô hình 15](#_Toc162778540)

[3.3. Giao diện 16](#_Toc162778541)

[TỔNG KẾT 16](#_Toc162778542)

[Nguồn tài liệu tham khảo: 17](#_Toc162778543)

**DANH MỤC CÁC HÌNH ẢNH**

[Hình 1 : Hình logo Đại Học Thủy Lợi 1](#_Toc162778557)

[Hình 2 : Tỷ lệ thiếu dữ liệu của các cột 5](#_Toc162778558)

[Hình 3: Giá trị trung bình của các cột bị thiếu 5](#_Toc162778559)

[Hình 4: Giá trị phương sai của các cột bị thiếu 6](#_Toc162778560)

[Hình 5: Sử dụng thư viện Label Encoder 6](#_Toc162778561)

[Hình 6: Biểu đồ box plot của cột B365H 7](#_Toc162778562)

[Hình 7: Kết quả sau khi xử lý dữ liệu nhiễu 8](#_Toc162778563)

[Hình 8: Xử lý dữ liệu nhiễu của cột B365D 8](#_Toc162778564)

[Hình 9: Xử lý dữ liệu nhiễu của cột B365A 9](#_Toc162778565)

[Hình 10: Xử lý dữ liệu nhiễu cột BSH 9](#_Toc162778566)

[Hình 11: Xử lý dữ liệu nhiễu cột BSD 9](#_Toc162778567)

[Hình 12: Xử lý dữ liệu nhiễu cột BSA 10](#_Toc162778568)

[Hình 13: Giá trị hỗn hợp 10](#_Toc162778569)

[Hình 14: Các bước xử lý giá trị hỗn hợp 11](#_Toc162778570)

[Hình 15: Các cột sau khi xử lý giá trị hỗn hợp 11](#_Toc162778571)

[Hình 16: Dữ liệu mất cân bằng trên cột Targe 12](#_Toc162778572)

[Hình 17: Dữ liệu sau khi được xử lý của cột target 13](#_Toc162778573)

[Hình 18: Sử dụng Standardization 14](#_Toc162778574)

# PHẦN 1. GIỚI THIỆU BÀI TOÁN

Trong bài toán này, chúng ta xem xét việc dự đoán kết quả các trận đấu bóng đá dựa trên bộ dữ liệu thực tế.

Dữ liệu cung cấp cho chúng ta thông tin về các trận đấu đã diễn ra tại giải đấu Europe và tỷ lệ cược trên các sàn cá độ tài trợ như Bet365 và Blue Square. Mục tiêu của dự án này là xây dựng mô hình học máy để phân loại các trận đấu thắng và thua trong giải đấu để cung cấp thông tin hợp lý cho các nhà cái dựa trên các thuộc tính đã có sẵn trong bộ dữ liệu.

**Dataset Information**

Tập dữ liệu này chứa thông tin về các đội bóng thuộc liên đoàn bóng đá Châu Âu và các trận đấu đã thi đấu cùng với tỉ lệ cược trên các sàn cá độ được diễn ra trong giải đấu Europe.

**Data Features**

Tập dữ liệu bao gồm 18 biến, bao gồm :

1. **ID:** ID của từng trận đấu
2. **Country\_name:** Tên quốc gia của đội bóng tham gia giải đấu
3. **League\_name:** Tên giải đấu
4. **Season:** Mùa giải thi đấu
5. **Date:** Ngày diễn ra trận đấu
6. **Home\_team:** Đội nhà
7. **Away\_team:** Đội khách
8. **B365H , B365D , B365A:** Tỷ lệ cá cược đội nhà thắng , hòa và đội khách thắng trên sàn 365bet
9. **BSH , BSD , BSA:** Tỷ lệ cá cược đội nhà thắng , hòa và đội khách thắng trên sàn Blue Square
10. **Diff\_goals:** Hiệu số bàn thắng
11. **Target:** Mục tiêu

**Data Preprocessing**

1. **Data Cleaning:** Kiểm tra các giá trị còn thiếu và xử lý chúng một cách thích hợp. Đảm bảo tính nhất quán trong các kiểu dữ liệu.
2. **Data Exploration:** Thực hiện phân tích dữ liệu khám phá (EDA) để hiểu sự phân bổ dữ liệu và mối quan hệ giữa các tính năng.
3. **Feature Engineering:** Create new features if necessary and encode categorical variables (e.g., one-hot encoding for education and marriage).
4. **Data Split:** Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá mô hình.

**Model Building**

1. **Model Selection:** Chọn một thuật toán học máy thích hợp để phân loại nhị phân. Các lựa chọn phổ biến bao gồm Logistic Regression
2. **Model Training:** Huấn luyện mô hình đã chọn trên dữ liệu huấn luyện.
3. **Model Evaluation:** Đánh giá hiệu suất của mô hình bằng cách sử dụng các số liệu thích hợp như độ chính xác, độ chính xác, khả năng thu hồi, điểm F1 trên dữ liệu thử nghiệm.

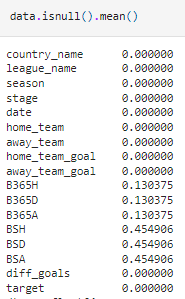
**Conclusion**

Dự án phân loại các trận đấu bao gồm quá trình tiền xử lý dữ liệu, xây dựng mô hình và triển khai mô hình học máy để phân loại các trận đấu bóng đá.

Dự án nhằm mục đích hỗ trợ cung cấp thông tin cho các nhà cái nhằm thao tác nhanh hơn và tổng hợp dữ liệu để đưa ra các phương án hợp lý cho các nhà cái.

# PHẦN 2. TIỀN XỬ LÝ DỮ LIỆU

## 2.1. Xử lý dữ liệu thiếu



Hình 2 : Tỷ lệ thiếu dữ liệu của các cột

Tỷ lệ các thông tin thiếu trong các cột dữ liệu:

B365H: 13%

B365D: 13%

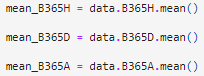
B365A: 13%

BSH: 45%

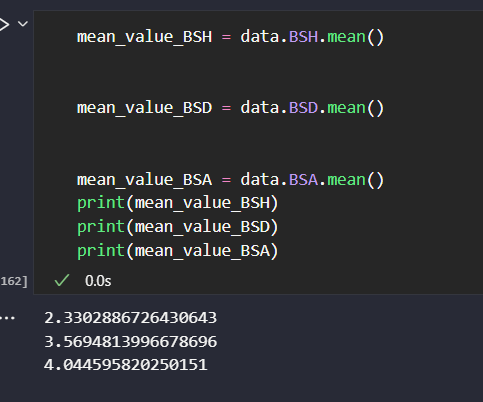
BSD: 45%

BSA: 45

Đối với các cột B365H,B365D,B365A tiến hành lấy giá trị trung bình của các giá trị trong cột đó qua hàm mean() .



Hình 3: Giá trị trung bình của các cột bị thiếu

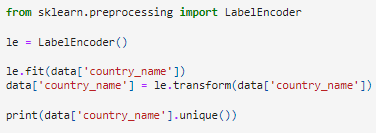


Hình 4: Giá trị phương sai của các cột bị thiếu

## 2.2. Mã hóa dữ liệu

Tiến hành sử dụng kĩ thuật Label Encoding mã hóa các cột các thông tin có dạng chuỗi chuyền về dạng chữ số.

Tiến hành mã hóa với cột country\_name.



Hình 5: Sử dụng thư viện Label Encoder

country\_name: có 10 nhãn

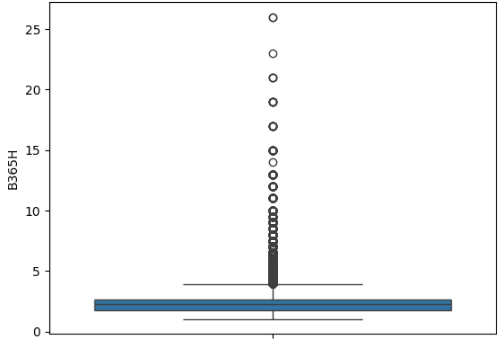


Kết quả:



Làm tương tự với các cột league\_name ,home\_team và away\_team.

## 2.3. Xử lý dữ liệu nhiễu



Hình 6: Biểu đồ box plot của cột B365H

Biểu đồ boxplot của cột B365H, ta có thể thấy nhiều dữ liệu nhiễu có giá trị bất thường không tuân theo nguyên tắc dữ liệu, nếu chạy mô hình dẫn đến kết quả không được tốt.

Kỹ thuật IQR (Interquartile Range) là một phương pháp phổ biến được sử dụng để xử lý dữ liệu nhiễu trong thống kê và khoa học dữ liệu. IQR đo độ phân tán của dữ liệu bằng cách tính sự khác biệt giữa phân vị thứ 75 (Q3) và phân vị thứ 25 (Q1). Kỹ thuật này thường được áp dụng trong việc xác định và loại bỏ các điểm dữ liệu nhiễu từ tập dữ liệu.

Các bước cụ thể để sử dụng kỹ thuật IQR để xử lý dữ liệu nhiễu là:

**Xác định IQR:** Tính IQR bằng cách lấy phân vị thứ 75 (Q3) trừ đi phân vị thứ 25 (Q1).



**Xác định ngưỡng cắt:** Xác định ngưỡng cắt cho dữ liệu nhiễu bằng cách nhân IQR với một hằng số cố định. Một cách phổ biến là sử dụng ngưỡng là *k*×*IQR*, trong đó *k* thường được đặt là 1.5 hoặc 3.

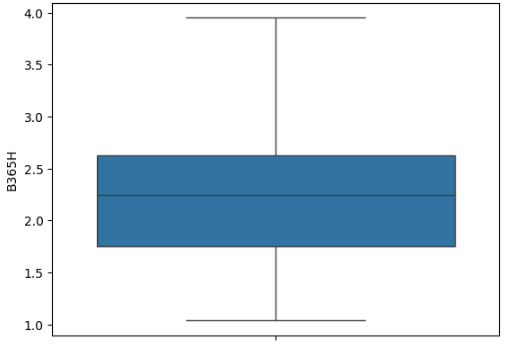


**Loại bỏ dữ liệu nhiễu:** Loại bỏ các điểm dữ liệu nằm ngoài phạm vi giới hạn được xác định bởi ngưỡng cắt.

**Phân tích hoặc thay thế dữ liệu nhiễu:** Sau khi loại bỏ dữ liệu nhiễu, bạn có thể tiến hành phân tích dữ liệu hoặc thực hiện các phương pháp khác như thay thế dữ liệu nhiễu bằng giá trị trung bình, trung vị hoặc dự đoán từ mô hình.



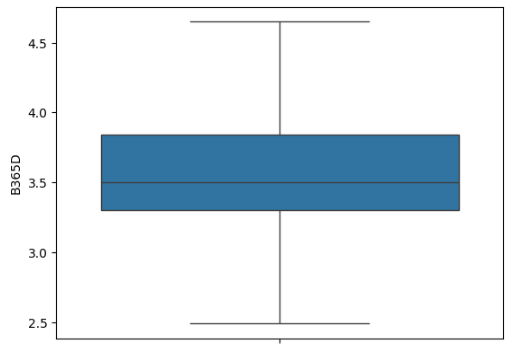
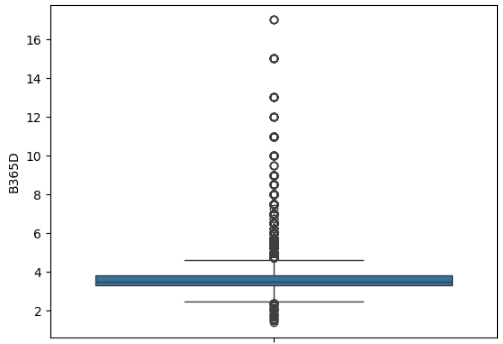
Kết quả:



Hình 7: Kết quả sau khi xử lý dữ liệu nhiễu

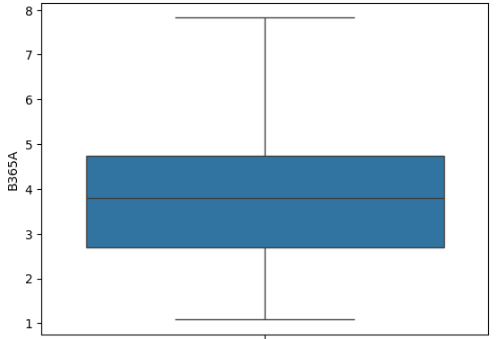
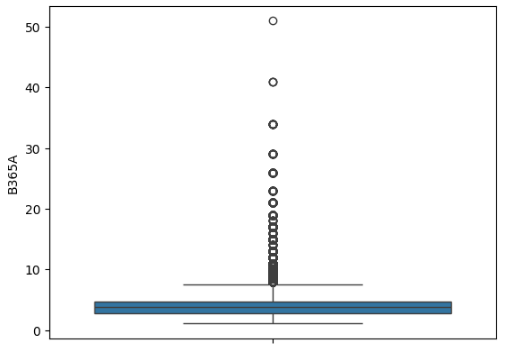
Làm tương tự với các cột còn lại.

B365D



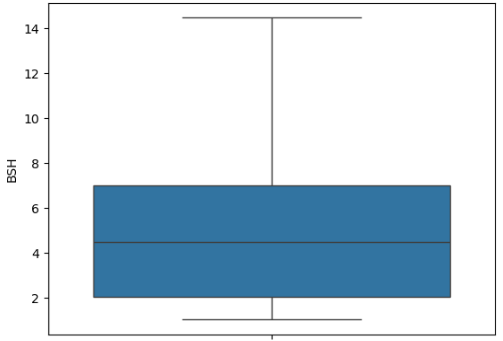
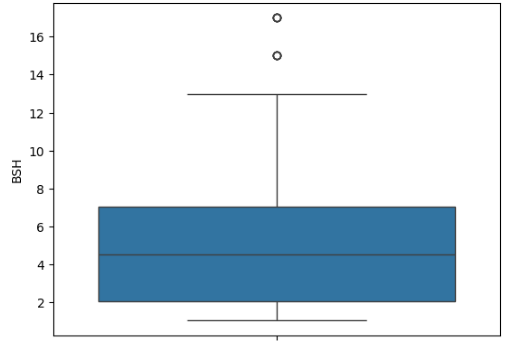
Hình 8: Xử lý dữ liệu nhiễu của cột B365D

B365A



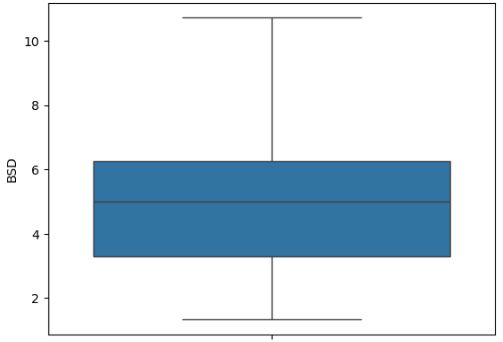
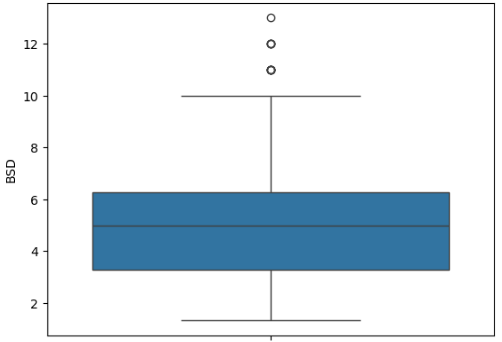
Hình 9: Xử lý dữ liệu nhiễu của cột B365A

BSH



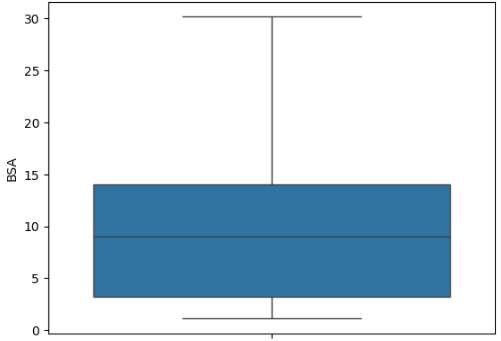
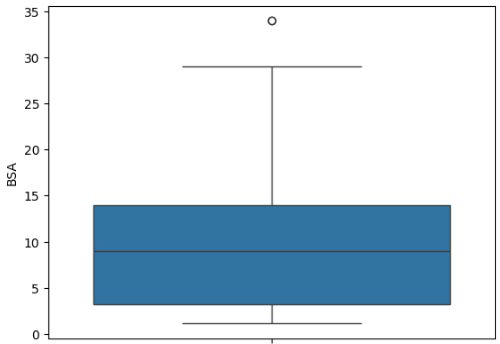
Hình 10: Xử lý dữ liệu nhiễu cột BSH

BSD



Hình 11: Xử lý dữ liệu nhiễu cột BSD

BSA



Hình 12: Xử lý dữ liệu nhiễu cột BSA

## 2.4. Xử lý giá trị hỗn hợp

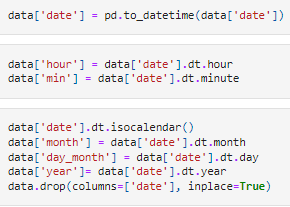


Hình 13: Giá trị hỗn hợp

Cột date có dữ liệu hỗn hợp giữa ngày, giờ và năm.

Trong các tập dữ liệu thực tế, có thể có trường hợp một cột chứa cả các giá trị số và các chuỗi văn bản. Điều này có thể gây ra khó khăn trong việc phân tích và xử lý dữ liệu, vì các phương pháp thống kê và học máy thường chỉ áp dụng cho một loại dữ liệu cụ thể.

Tiến hành phân tách thành các cột riêng biệt: mỗi cột đại diện cho một loại giá trị cụ thể (văn bản hoặc số). Sau đó, có thể áp dụng các phương pháp xử lý dữ liệu tương ứng cho mỗi loại giá trị.



Hình 14: Các bước xử lý giá trị hỗn hợp

Tiến hành chia thời gian hỗn hợp thành nhiều cột riêng biệt về giá trị.

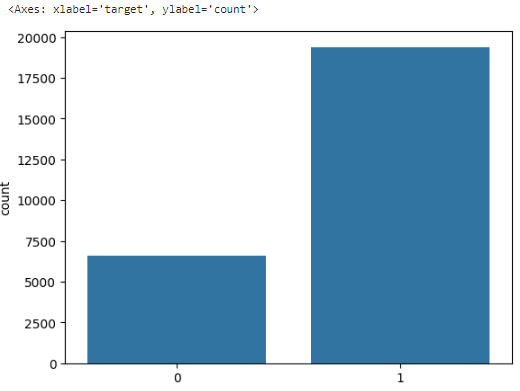
Tương tự, chia thời gian của season thành 2 cột start\_season và end\_season.

Kết quả:



Hình 15: Các cột sau khi xử lý giá trị hỗn hợp

## 2.5. Xử lý mất cân bằng dữ liệu



Hình 16: Dữ liệu mất cân bằng trên cột Targe

Cột target có số lượng không cân bằng giữa các nhãn.

Trong một số bài toán phân loại, có thể xảy ra trường hợp một nhóm lớp có số lượng quan sát ít hơn đáng kể so với các nhóm khác, điều này có thể làm cho mô hình học máy không hoạt động hiệu quả, vì mô hình sẽ có xu hướng học cách dự đoán nhãn của lớp đa số.

Dưới đây là một số phương pháp phổ biến để xử lý mất cân bằng dữ liệu:

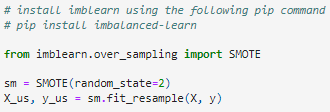
Over-sampling (Tăng cường mẫu): Phương pháp này tạo ra thêm mẫu cho các lớp thiểu số bằng cách lặp lại các mẫu hiện có hoặc tạo ra các mẫu mới dựa trên dữ liệu hiện có.

Under-sampling (Giảm mẫu): Phương pháp này loại bỏ một số mẫu từ các lớp đa số để làm cho tỷ lệ giữa các lớp cân bằng hơn. Điều này có thể dẫn đến mất mát thông tin nếu không thực hiện cẩn thận.

Trong bài toán ta tiến hành chọn phương pháp over-sampling.



Lấy ra các cột thuộc tính và cột nhãn để tiến hành phương pháp.



SMOTE là một kỹ thuật phổ biến giúp tăng cường số lượng mẫu trong các lớp thiểu số bằng cách tạo ra các mẫu tổng hợp.

Cách SMOTE hoạt động như sau:

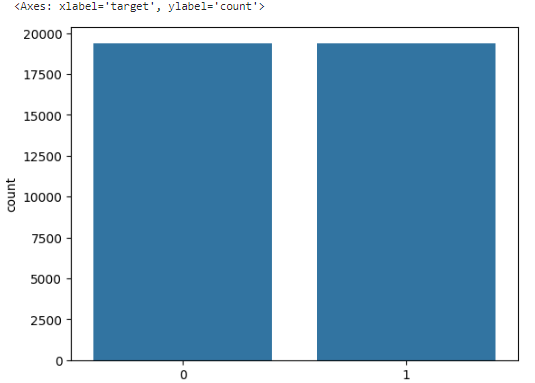
**Chọn mẫu cơ sở:** Đầu tiên, chúng ta chọn một mẫu từ lớp thiểu số (nhãn hiếm) trong tập dữ liệu.

**Tìm hàng xóm gần kề:** Sau đó, chúng ta sẽ tìm các hàng xóm gần kề của mẫu đã chọn từ lớp thiểu số, thường dùng k-nearest neighbors (k-NN).

**Tạo mẫu tổng hợp:** Cho mỗi mẫu cơ sở, chúng ta chọn một trong các hàng xóm gần kề và tạo ra một mẫu tổng hợp mới.

**Thêm mẫu mới vào tập dữ liệu:** Mẫu tổng hợp mới được thêm vào tập dữ liệu, làm tăng số lượng mẫu trong lớp thiểu số.

Kết quả:

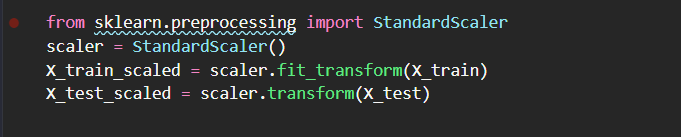


Hình 17: Dữ liệu sau khi được xử lý của cột target

## 2.6. Feature Scaling

Feature scaling là quá trình điều chỉnh tỷ lệ của các biến đặc trưng trong tập dữ liệu sao cho chúng có cùng phạm vi hoặc đơn vị đo. Mục tiêu của việc làm này là để đảm bảo rằng các biến đặc trưng có ảnh hưởng tương đồng đối với quá trình học của mô hình học máy hoặc thuật toán khác

Standardization (Z-score Normalization): Là phương pháp chuyển đổi các giá trị của biến sao cho chúng có giá trị trung bình gần bằng 0 và độ lệch chuẩn (standard deviation) gần bằngd 1. Công thức cho phép tính giá trị sau chuẩn hóa của một biến là: Xscaled =



Hình 18: Sử dụng Standardization

# PHẦN 3. ỨNG DỤNG MÔ HÌNH HỌC MÁY ĐỂ THỰC HIỆN VỚI BÀI TOÁN

Sau khi dữ liệu đã được tiền xử lý, ta có thể tiến hành xây dựng mô hình học máy cho bài toán.

Cụ thể, các bước tiến hành như sau:

* Chia dữ liệu thành tập train và test để huấn luyện mô hình.
* Chạy mô hình bằng tập train.
* Kiểm tra độ chính xác của mô hình bằng tập test và các độ đo như F1,recall,…

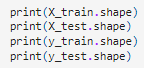
## 3.1. Chia dữ liệu

****

Sử dụng hàm train\_test\_split chia tập dữ liệu thành 2 phần trong đó:

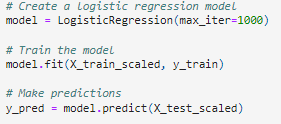
* Tập Ttrain chiếm 70%
* Tập Test chiếm 30%

Kết quả:



## 3.2. Huấn luyện mô hình

Chọn mô hình Logistic Regression.

****

**Hàm fit:** huấn luyện mô hình theo tập train.

**Hàm predict:** đưa ra các giá trị dự đoán dựa trên tập test.

## 3.3. Kiểm tra độ chính xác của mô hình



Sử dụng 2 độ đo là accuracy và report.

**Accuracy (Độ chính xác):**

Accuracy là tỷ lệ phần trăm giữa số lượng dự đoán đúng và tổng số lượng mẫu trong tập dữ liệu kiểm tra.

Công thức tính: Accuracy = (Số lượng dự đoán đúng) / (Tổng số lượng mẫu)

Độ chính xác càng cao thì mô hình càng tốt. Tuy nhiên, độ chính xác không phản ánh được hiệu suất của mô hình trong trường hợp dữ liệu mất cân bằng hoặc khi có các lớp có tỷ lệ khác nhau.

**Classification Report (Báo cáo phân loại):**

Classification report là một báo cáo chi tiết về hiệu suất của mô hình trên các lớp (classes) khác nhau trong bài toán phân loại.

Bao gồm các thông số như precision, recall, F1-score và support cho mỗi lớp.

Precision (độ chính xác): Tỷ lệ giữa số lượng dự đoán đúng của lớp đó và tổng số lượng mẫu được dự đoán là lớp đó.

Recall (độ nhớ lại): Tỷ lệ giữa số lượng dự đoán đúng của lớp đó và tổng số lượng mẫu thuộc lớp đó trong dữ liệu thực tế.

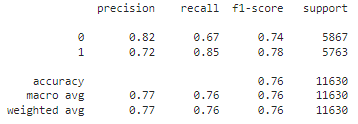
F1-score: Trung bình điều hòa của precision và recall, cung cấp một phép đo tổng thể về hiệu suất của mô hình.

Support: Số lượng mẫu thực tế thuộc lớp đó.

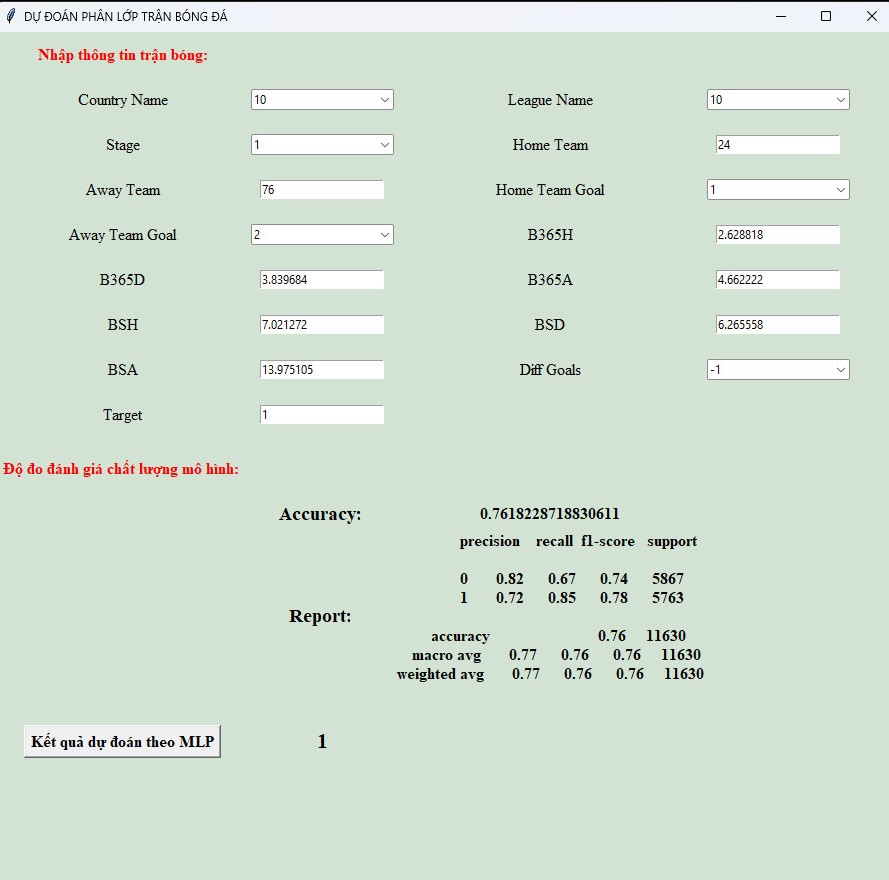
Kết quả:

**Accuracy:** 0.76

**Classification Report:**



## 3.3. Giao diện



Hình 19: Giao diện mô hình học máy

# TỔNG KẾT

* + Trong bài báo cáo này, chúng em đã thực hiện một loạt các bước tiền xử lý dữ liệu và chạy mô hình học máy để giải quyết bài toán. Chúng em đã đạt được một số kết quả quan trọng và nhận định sau đây:

**Tiền xử lý dữ liệu:**

* Chúng em đã thu thập và tiền xử lý một tập dữ liệu để chuẩn bị cho quá trình huấn luyện mô hình.
* Các phương pháp tiền xử lý đã được áp dụng để giải quyết các vấn đề ban đầu của dữ liệu.

**Chạy mô hình học máy:**

* Chúng em đã chọn và huấn luyện một số mô hình học máy trên tập dữ liệu đã tiền xử lý.
* Qua quá trình huấn luyện, mô hình cho kết quả tương đối ổn định trên các thang đo như recall, f1 score,…

**Nhận xét và kết luận:**

* Từ quá trình tiền xử lý dữ liệu và chạy mô hình học máy, chúng em nhận thấy như sau:

**Uu điểm:**

* Các phương pháp tiền xử lý khá nhanh và hiệu quả.
* Có thể thay đổi nhiều phương pháp để linh hoạt tùy vào bài toán.

**Hạn chế:**

* Có thể một số phương pháp gây ra mất thông tin đặc trưng về dữ liệu.
* Nhiều phương pháp có thể tốn nhiều thời gian và không hiệu quả khi dữ liệu không phù hợp.

Chúng em cũng xin đề xuất một vài ý tưởng:

* Nghiên cứu và thử nghiệm các phương pháp tiền xử lý dữ liệu mới để giảm thiểu mất mát thông tin và tăng cường khả năng tổng quát hóa của mô hình.
* Áp dụng các kỹ thuật tiền xử lý dữ liệu tiên tiến như feature engineering để tạo ra các đặc trưng mới có thể cải thiện hiệu suất của mô hình.

Nhìn chung, quá trình này đã cung cấp cái nhìn sâu rộng về quá trình tiền xử lý dữ liệu và chạy mô hình học máy trong việc giải quyết bài toán thực tế. Những kết quả và nhận định từ bài toán này có thể hữu ích cho các nghiên cứu và ứng dụng trong tương lai.

# Nguồn tài liệu tham khảo:

**Lý thuyết:**

* Slide bài giảng của TS. Tạ Quang Chiểu, đại học Thủy Lợi.
* Machinelearningcoban.com
* Book data Preprocessing Python master